Style Breach Detection with Neural Sentence Embeddings

Notebook for PAN at CLEF 2017

Kamil Safin, Rita Kuznetsova

Antiplagiat CJSC,
Moscow Institute of Physics and Technology,
safin@ap-team.ru, kuznetsova@ap-team.ru

Abstract The paper investigates method for the style breach detection task. We developed a method based on mapping sentences into high dimensional vector space. Each sentence vector depends on the previous and next sentence vectors. As main architecture for this mapping we use the pre-trained encoder-decoder model. Then we use these vectors for constructing an author style function and detecting outliers. Method was tested on the PAN-2017 collection for the style breach detection task.

1 Introduction

Developing approach for identifying different authors within a single document has been an open problem at the natural language processing. There were several tasks related to this problem in PAN competition:

1. Intrinsic plagiarism detection problem [10,14,7] — given a suspicious document that there exists one main author who wrote at least 70% of the text. Up to the other 30% may be written by other authors. The task is to determine whether the document is written by a single author or contains fragments by another authors. Unlike external plagiarism problem, the reference collection is unknown [16].
2. Author diarization problem [13] — given the document, that written by n authors, no main author is given. The task is to determine exactly n authors in the document, where the number n can be known or unknown.

The most algorithm's work is based on the following scheme:

1. divide a text into blocks according to the segmentation scheme (e.g. sentences, n -grams, overlapping blocks),
2. map each block to feature space (e.g. n-gram frequency [1,12], punctuation, part-of-speech tags count [5]) and combine features to an author style function (character 3 -gram frequencies, n -gram classes (i.e. the inverted frequencies), normalized word frequency class),
3. find critical values in the author style function to detect plagiarized blocks. The author diarization algorithms [4] use segmentation of classifier statistics if the number of authors is known and the clustering approach if the the number of authors is unknown.

PAN -2017 [9] competition provided modified problem statement — style breach detection [15]. Given a document, determine whether it is multi-authored, and if yes, find the borders where authors switch. For this task we proposed the approach based on neural phrase embeddings. First, we split a document into sentences and map each sentences in high dimensional vector space using pretrained encoder-decoder model named skip-thoughts model from [3]. Each sentence vector depends on the sentence vector before and after it. After that, we construct the similarity matrix between all sentences in document and detect outliers.
The quality of the model was measured by $WindowDiff$ [6] and $WinP, WinR, WinF$ [11] metrics. All experiments were carried out on TIRA [8].

2 Style Breach Detection

Denote D the collection of text documents. Each document $d \in D$ is written by unknown number of authors. The task is to find borders where authors switch. All documents may contain zero up to arbitrarily many switches. Thereby switches of authorship may only occur at the end of sentences, i.e. not within.

We formulate style breach detection problem as finding sentences-outliers problem. Text document $d \in D$ consists of sentences: $d = \cup_{i=1}^{N} s_i$, where N — number of sentences in text. Each of sentences s_i we vectorize, using pre-trained skip-thoughts model: $s_i \rightarrow \mathbf{s}_i$. Then, statistic for sentences $stat(\mathbf{s}_i)$ is built, and the problem is to find sentences, which statistic is bigger than statistic of other sentences, in other words, the goal is to find sentence vectors, which statistic is exceeded the threshold:

$$stat(\mathbf{s}_i) > \delta \Rightarrow s_i \; is \; outlier.$$

3 Experiment

3.1 Quality criteria

To evaluate the predicted style breaches two metrics were used:

– WindowDiff metric was proposed for general text segmentation evaluation. It gives an error rate (between 0 to 1, where 0 indicates a perfect prediction) for predicting borders by penalizing near-misses less than other/complete misses or extra borders. This metric computes as follows:

$$WindowDiff(ref, hyp) = \frac{1}{N-k} \sum_{i=1}^{N-k} (|b(ref_i, ref_{i+k}) - b(hyp_i, hyp_{i+k})| > 0),$$

where $b(i, j)$ represents the number of boundaries between positions i and j in the text and N represents the number of sentences in the text, ref and hyp are reference and hypothetical segmentations.

– a more recent adaption of WindowDiff metric is WinPR metric. It enhances it by computing the common information retrieval measures precision (WinP) and recall (WinR) and thus allows to give a more detailed, qualitative statement about the prediction.

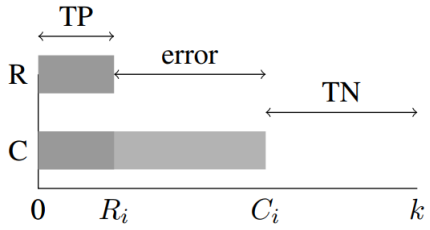
$$True\ Positives = TP = \sum_{i=1-k}^{N} \min(R_{i,i+k}, C_{i,i+k}),$$

$$True\ Negatives = TN = -k(k-1) + \sum_{i=1-k}^{N} (k - \max(R_{i,i+k}, C_{i,i+k})),$$

$$False\ Positives = FP = \sum_{i=1-k}^{N} \max(0, C_{i,i+k} - R_{i,i+k}),$$

$$False\ Negatives = FN = \sum_{i=1-k}^{N} \max(0, R_{i,i+k} - C_{i,i+k}),$$

where R and C represent the number of boundaries from the reference and computed segmentations, respectively, in the i^{th} window, up to a maximum of k; N is the number of content units and k represents the window size.



And WinP, WinR, WinF are computed as:

$$WinP = \frac{TP}{TP + FP},$$

$$WinR = \frac{TP}{TP + FN},$$

$$WinF = \frac{2 \cdot WinP \cdot WinR}{WinP + WinR}$$

3.2 Feature construction

The raw text document d is splitted into sentences s_i using standart NLTK's sentence tokenizer [2]. Each sentence is vectorized by pre-trained skip-thoughts model[1]. Skip-thoughts model belongs to the class of encoder-decoder models. That is, encoder part

[1] `https://github.com/ryankiros/skip-thoughts`

maps word embeddings to a sentence vector and decoder generates surrounding sentences. Skip-thought vectors consist of two separate models. One is an unidirectional encoder with 2400 dimensions, which is referred to as uni-skip. The other is a bidirectional model with 2400 dimensions, that contains forward and backward encoders of 1200 dimensions each. This model is referred to as bi-skip.

Encoder. Let $w_i^1, \ldots, w_i^N$ be the words in sentence s_i and N is the number of words in sentence. On each step, encoder generates hidden state $\mathbf{h}_i^t$, which can be interpreted as the representation of the sequence $w_i^1, \ldots, w_i^t$. And the final hidden state $\mathbf{h}_i^N := \mathbf{s}_i$ is the vector representation of the full sentence s_i.

$$\begin{aligned} \mathbf{z}_t &= \sigma(\mathbf{W}_{zx}\mathbf{x}_t + \mathbf{W}_{zh}\mathbf{h}_{t-1}), \\ \mathbf{r}_t &= \sigma(\mathbf{W}_{rx}\mathbf{x}_t + \mathbf{W}_{rh}\mathbf{h}_{t-1}), \\ \tilde{\mathbf{h}}_t &= \tanh(\mathbf{W}_x\mathbf{x}_t + \mathbf{W}_h(\mathbf{r}_t \circ \mathbf{h}_{t-1})), \\ \mathbf{h}_t &= (1 - \mathbf{z}_t) \circ \mathbf{h}_{t-1} + \mathbf{z}_t \circ \tilde{\mathbf{h}}_t, \end{aligned} \tag{1}$$

where $(\mathbf{W}_{zx}, \mathbf{W}_{zh}, \mathbf{W}_{rx}, \mathbf{W}_{rh}, \mathbf{W}_x, \mathbf{W}_h)$ — parameters of LSTM type encoder, $\mathbf{x}_t$ — vector representation of word w_t, $(\circ)$ denotes a component-wise product.

Decoder. The decoder is a model which conditions on the encoder output $\mathbf{s}_i$. Decoder part is similar to encoder part, but applied to next s_{i+1} and previous s_{i-1} sentences.

Objective. Given a tuple (s_{i-1}, s_i, s_{i+1}) the objective optimized is the sum of the log-probabilities for the forward and backward sentences conditioned on the encoder representation.

Consider the dataset $S = \{s_i\}$ consisting of the sentences $s_i = (\mathbf{x}_1, \ldots, \mathbf{x}_n)$ where $\mathbf{x}_k \in \mathbf{X}$ is a word embedding. Our goal is to learn representations for variable-sized phrases in unsupervised training regime. We use the encoder-decoder model (GRU-GRU) described in [3].

To build statistics, we construct pairwise distance matrix $M = \{m_{ij}\}_{i,j=1}^N$, where N is the number sentences in text. For each pair of sentences $(\mathbf{s}_i, \mathbf{s}_j)$ cosine distance is computed:

$$m_{ij} = \cos(\mathbf{s}_i, \mathbf{s}_j).$$

Statistic for each sentence is built as mean cosine distance to all other sentences in text:

$$stat(\mathbf{s}_i) = \frac{1}{N} \sum_{j \neq i} \cos(\mathbf{s}_i, \mathbf{s}_j).$$

To detect borders, where authors switch, we accept the hypothesis, that sentences around the borders are differ from other sentences in text. Outliers are defined as sentences, which statistic is bigger than threshold δ:

$$stat(\mathbf{s}_i) > \delta \Rightarrow s_i \; is \; outlier.$$

The example of work of the algorithm is shown below. Green line denotes threshold value, red lines mark detected sentences-outliers. Blue dots on pairwise distance matrix denote real borders.

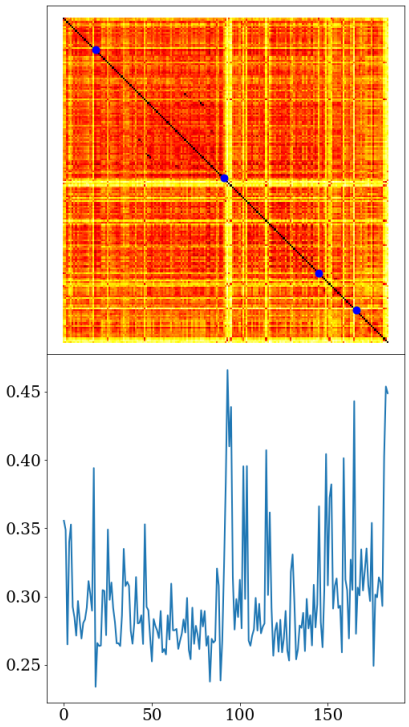


Figure 1: Example of pairwise distance matrix and statistic for sentences

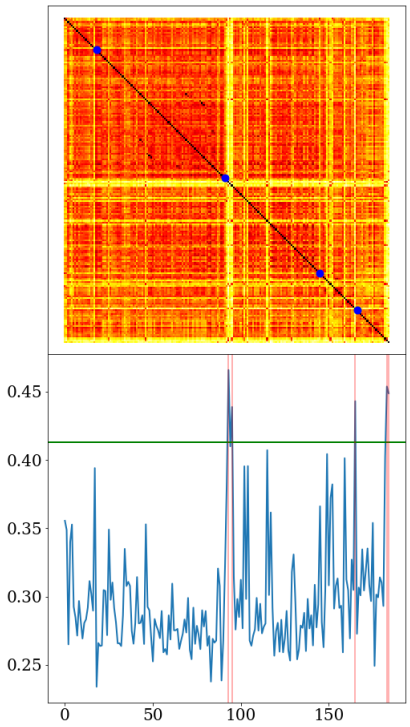


Figure 2: Pairwise distance matrix and statistic for sentences with threshold(green) and detected outliers(red)

3.3 Parameters Tuning

The threshold δ was tuned in order to maximize the final performance measure — $WinF$. Also, to compress model and analyze the properties of skip-thoughts vectors, different parts of these vectors were used for statistic calculations, specifically:

- whole 4800-dimensional skip-thoughts vectors,
- 2400-dimensional uni-skip vectors,
- 2400-dimensional bi-skip vectors.

The results of parameter tuning are shown on figures below.

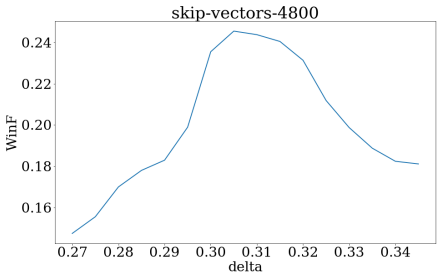


Figure 3: Skip-vectors model parameters tuning.

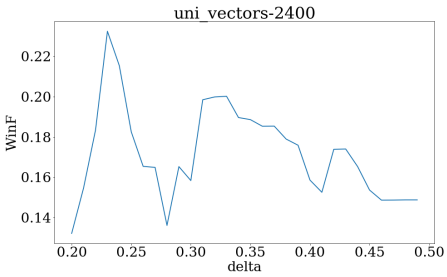


Figure 4: Uni-vectors model parameters tuning.

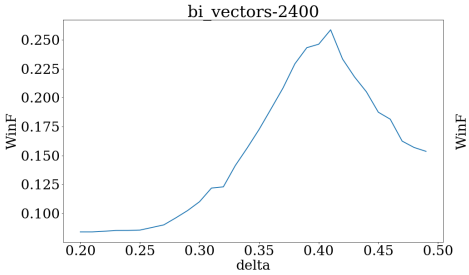


Figure 5: Bi-vectors model parameters tuning.

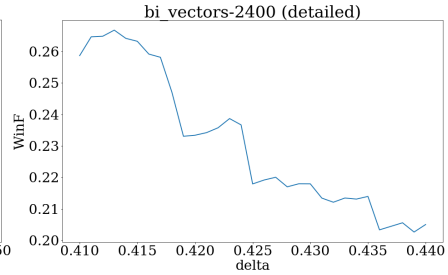


Figure 6: Uni-vectors model precise parameters tuning.

3.4 Results

The proposed algorithm was tested on PAN-2017 style breach detection training and test datasets. Results of its work are shown in table below.

	WindowDiff	WinP	WinR	WinF
training dataset	0.62	0.27	0.61	**0.24**
test dataset	0.53	0.37	0.54	**0.28**

Table 1: Results on PAN'17 data set

4 Conclusion

We proposed algorithm for style breach detection task. This method splits text into sentences, vectorizes it and then builds statistics for sentence vectors to detect sentences-outliers.
The method was implemented to the PAN-2017 competition in style breach detection task. The model achieved WinF measure 0.28 on the test dataset.

References

1. Bensalem, I., Rosso, P., Chikhi, S.: Intrinsic plagiarism detection using n-gram classes. EMNLP (2014)
2. Bird, S.: Nltk: the natural language toolkit. Proceedings of the COLING/ACL on Interactive presentation sessions (2006)
3. Kiros, R., Zhu, Y., Salakhutdinov, R., Zemel, R.S., Torralba, A., Urtasun, R., Fidler, S.: Skip-thought vectors. arXiv preprint arXiv:1506.06726 (2015)
4. Kuznetsov, M., Motrenko, A., Kuznetsova, R., Strijov, V.: Methods for intrinsic plagiarism detection and author diarization. Notebook for PAN at CLEF 2016 (2016)
5. Oberreuter, G., L'Huillier, G., Ríos, S.A., Velásquez, J.D.: Approaches for intrinsic and external plagiarism detection. Proceedings of the PAN (2011)
6. Pevzner, L., Hearst, M.A.: A critique and improvement of an evaluation metric for text segmentation. Computational Linguistics (2002)
7. Potthast, M., Gollub, T., Hagen, M., Kiesel, J., Michel, M., Oberländer, A., Tippmann, M., Barrón-Cedeño, A., Gupta, P., Rosso, P., Stein, B.: Overview of the 4th international competition on plagiarism detection. CLEF (Online Working Notes/Labs/Workshop). Citeseer (2012)
8. Potthast, M., Gollub, T., Rangel, F., Rosso, P., Stamatatos, E., Stein, B.: Improving the Reproducibility of PAN's Shared Tasks: Plagiarism Detection, Author Identification, and Author Profiling. In: Kanoulas, E., Lupu, M., Clough, P., Sanderson, M., Hall, M., Hanbury, A., Toms, E. (eds.) Information Access Evaluation meets Multilinguality, Multimodality, and Visualization. 5th International Conference of the CLEF Initiative (CLEF 14). pp. 268–299. Springer, Berlin Heidelberg New York (Sep 2014)
9. Potthast, M., Rangel, F., Tschuggnall, M., Stamatatos, E., Rosso, P., Stein, B.: Overview of PAN'17: Author Identification, Author Profiling, and Author Obfuscation. In: Jones, G., Lawless, S., Gonzalo, J., Kelly, L., Goeuriot, L., Mandl, T., Cappellato, L., Ferro, N. (eds.) Experimental IR Meets Multilinguality, Multimodality, and Interaction. 8th International Conference of the CLEF Initiative (CLEF 17). Springer, Berlin Heidelberg New York (Sep 2017)
10. Potthast, M., Stein, B., Barrón-Cedeño, A., Rosso, P.: An evaluation framework for plagiarism detection. Proceedings of the 23rd international conference on computational linguistics (2010)
11. Scaiano, M., Inkpen, D.: Getting more from segmentation evaluation. Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (2012)
12. Stamatatos, E.: Intrinsic plagiarism detection using character n-gram profiles (2009)
13. Stamatatos, E., Tschuggnall, M., Verhoeven, B., Daelemans, W., Specht, G., Stein, B., Potthast, M.: Clustering by authorship within and across documents. CEUR Workshop Proceedings (2016)
14. Stein, B., Barron, Cedeno, L., Eiselt, A., Potthast, M., Rosso, P.: Overview of the 3rd international competition on plagiarism detection. CEUR Workshop Proceedings (2011)
15. Tschuggnall, M., Stamatatos, E., Verhoeven, B., Daelemans, W., Specht, G., Stein, B., Potthast, M.: In: Cappellato, L., Ferro, N., Goeuriot, L., Mandl, T. (eds.) Working Notes Papers of the CLEF 2017 Evaluation Labs
16. Zechner, M., Muhr, M., Kern, R., Granitzer, M.: External and intrinsic plagiarism detection using vector space models. Proc. SEPLN. vol. 32 (2009)